印刷体汉字识别中极相似字区别的研究

中国科学院沈阳自动化研究所

梁刚　武裕朴

摘　要

本文通过对类似度原理的分析，论证和实验，实现了区别极相似字能力以及克服噪声干扰能力较强的混合类似度和克服噪声干扰能力较强的复合类似度，并将混合类似度纳入本印刷体汉字识别系统的最后一阶，获得较好的识别结果。另外，利用简单类似度对汉字库中3023个32×32的汉字模式进行了相似字的统计工作，建立了可供查询的相似汉字库，分析了图形汉字的相似结构，以便提供抽取汉字特征的依据。整个实验是在*PDP*－11/23机上进行的。

一、前　言

近几年来，在汉字识别方面已经做了大量的研究工作，这些工作可分为三大类：①印刷体汉字识别②在线手写体汉字识别③手写体汉字识别。就印刷体汉字识别而言，实验方法日渐成熟，尤其是日本在这方面取得了较好的成绩。如*Shouich Hira*等人研制的*KOCR*装置〔1〕，可识别2000个单字体汉字，其识别率为99.9%，识别速度可达每秒100字，并已实用。由于我国汉字的一些特点，以及研究工作起步较晚，使得印刷体汉字识别的研究还处于实验阶段。目前的特征抽取方法及识别方法较多，但怎样对这些方法进行改进并组合成一个完善的识别系统有待于进一步的研究。另外由于汉字的种类繁多，许多汉字在图形上都比较相似，而一般的识别方法又无法将这些图形上非常相似的汉字区分开，所以怎样区别极相似字也是有待于解决的问题。本文通过对类似度原理的分析、证明，实现了克服噪声干扰以及区别极相似字能力较强的混合类似度，并利用简单类似度对字库中3023个标准字样进行类似程度的统计工作，建立了相似汉字库。

二、类似度原理

类似度准则——类似度准则是表示待识文字图形与标准文字图形间相似程度的度量。

工作原理：候选集内每一个字符是用n维矩阵图形描述的，即$y_K=(y_{K1}, y_{K2}, \cdots y_{Kn})^T$ $K=1, 2, \cdots m$。式中m是候选集中的字符个数，在给定的图形向量$x=(x_1 \cdots x_n)^T$表述后，能计算如下定义的类似度函数 $S(x, y_K)=(x, y_K)/\sqrt{(x,x)(y_K, y_K)}=\cos\theta \cdots 2.1$。式中$(x, y_K)$为向量$x$与向量$y_K$的内积，具有最大类似度值的字种被指定为待识文字的种类。

上面提出的类似度函数是类似度技术中最简单的形式，称为简单类似度。这种方法的特点是计算简单，意义明确，但对于输入图形的位置变化以及随机噪声干扰的克服能力很差。为解决这些问题而引进了复合类似度。

设候选文字集为$\{f_0\}$，待识文字为g_0，根据$K-L$展开分析提供的文字图形的主要分布〔3〕，构造了三个相互正交的图形模式，并且它们包含了各种典型的噪声变化。如用处于标准位置的标准图形模式经纵向移动过的标准图形模式，经横向移动过的标准图形模式，组成三个相互正交的图形模式。当要求输入文字图形与某一种文字图形的复合类似度时，就是先分别求与此种文字图形三个相互正交的图形模式的简单类似度的平方和，其平方根即为复合类似度。

若三个相互正交的图形模式分别为φ_0、φ_1、φ_2，则

$$\varphi_0=g_0/\|g_0\| \qquad 2.2$$

$$\varphi_1=(g_1/\|g_1\|+g_2/\|g_2\|)/\sqrt{2(1+\theta)} \qquad 2.3$$

$$\varphi_2=(g_1/\|g_1\|-g_2/\|g_2\|)/\sqrt{2(1-\theta)} \qquad 2.4$$

其中 $g_1=\partial g_0/\partial x$ $g_2=\partial g_0/\partial y$ $\theta=(g_1,g_2)/\|g_1\|\cdot\|g_2\|$

复合类似度的计算公式为

$$S(f_0,g_0)=\sqrt{\sum_{j=0}^{2}(f_0,\varphi_j)^2}\Big/\|f_0\| \qquad 2.5$$

$$0\leqslant S(f_0,g_0)\leqslant 1$$

当 $f_0=g_0$ 时 $S(f_0,g_0)=1$

在识别时，取最大类似度值所对应的候选文字 l 为待识文字的种类，

即 $$S(f_0^l,g_0)=\max_k S(f_0^k,g_0) \qquad 1\leqslant l\leqslant k$$

以复合类似度为标准，对混有噪声的待识文字图形能稳定的识别，但对于非常相似的字如：推—惟；镜—境；苟—苟等却不能很好地区分。为克服上面缺点而引入混合类似度。

假定 f_0 为与 g_0 极相似的一文字的标准图形，则可定义两相似文字的差异图形 ψ

$$\psi=\Big[f_0-\sum_{j=0}^{2}(f_0,\phi_j)\phi_j\Big]\Big/\sqrt{\|f_0\|^2-\sum_{j=0}^{2}(f_0,\phi_j)^2} \qquad 2.6$$

ψ 为与 φ_j 的正交图形，即 $(\varphi_j,\psi)=0$ $j=0,1,2$

混合类似度 S^* 定义为

$$S^*=\sqrt{\sum_{j=0}^{2}(f_0,\varphi_j)^2-\mu(f_0,\psi)^2}\Big/\|f_0\| \qquad 2.7$$

其中 μ 为常数，可以由 $0\leqslant S^*\leqslant 1$ 推算确定。

三、用简单类似度进行相似字的统计

本文依据简单类似度的准则对汉字库中3023个标准文字图形进行了相似字的统计工作，并建立起相似汉字库。其目的是更多地了解极相似字的有关信息，并知道那些字是相似的，构成相似字的主要图形结构是什么样的。统计结果表明：

1。用此方法统计出来的极相似字都是汉字形体上非常相似的字，如：且—旦；士—土；推—惟；证—怔。

2。一般相似字为以下几种结构

1）结构上非常相似的独体字结构如：且，旦等；

2）汉字的50%是左右结构的字，而在这些字中，相似程度较大的是那些具有相同偏旁，且偏旁占文字笔式比重较大。如：悼—掉；塘—搪等；

3）半包体或全包体内部结构相似的字。如：冈—网；洞—涧等。

这里在相似字的统计中，对简单类似度值较大的相似汉字记录下来，并形成一相似汉字库。另外，利用本识别系统中所用的一些特征，象文字图形 x 轴、y 轴投影幅值的 R 变换系数特征，文字线长度特征〔5〕〔6〕做了类间可分性验证。发现许多相似字由这些特征就可以很好地区分开，具体反映在某些特征值上数值相差很大，这样，在某种程度上，我们说一些特征具有区分部分相似字的能力，当然我们希望，在特征容易抽取的前提下，应尽量提高区别相似字的能力。

四、复合类似度和混合类似度的识别算法及性能

前面我们已经提出复合类似度和混合类似度的原理，下面我们给出这两种方法的识别算法及实验结果。

混合类似度的识别算法

1。将待识文字图形 X 模糊化成 X_0

2。求出 X_0 的微分 $\partial X_0/\partial X$、$\partial X_0/\partial y$

3。制作三个相互正交的图形 φ_0、φ_1、φ_2

4。求候选文字y_i与待识文字的φ_0，φ_1，φ_2的内积，$i \leqslant K$，K为候选文字的个数

5。求差异图形ψ

6。计算混合类似度S_i^*

7。IF $i \leqslant K$ 转4

8。找出$S_i^{*l} = \max_k S_i^*$中的候选文字l为待识文字的种类

9。结束。

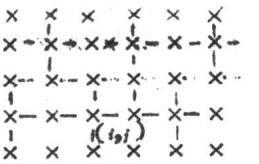

图 1

这里的模糊化过程可用图1来表示。

$$X_0(m, n) = X(i-1, j) + X(i, j-1) + X(i, j+1) + X(i+1, j) + X(i, j)$$

而X_0微分是选用3×3的梯度模板制成的。另外这里只给出了混合类似度的识别算法，复合类似度的识别算法差不多相同，只是没有步骤5，通过进一步利用相似汉字库对这两种方法的实验及调节参数，总结出这两种方法的以下几种性能：

① 克服位移和随机噪声干扰的能力强，表1给出了用混合类似度计算的带噪声的文字与它的标准文字本身之间的类似度、无噪声时$S^*(f_0, f_0) = 1$。

表1

噪声形式 \ 混合类似度值 \ 文字	她	愉	士	；	侧	概	盼	闷
位 移	0.897	0.916	0.850	0.909	0.947	0.906	0.869	0.864
随机噪声干扰	0.983	0.976	0.985	0.990	0.941	0.957	0.974	0.979

② 混合类似度区别极相似字的能力是极强的，表2给出了几种类似度的比较。

表2

方法 \ 类似度值 \ 相似字对	且，旦	朱，未	士，土	推，惟	刀，刃	闸，闸	浙，浙	官，官	；，：
简单类似度	0.946	0.819	0.899	0.886	0.854	0.845	0.840	0.821	0.866
复合类似度	0.949	0.918	0.918	0.929	0.926	0.916	0.926	0.904	0.909
混合类似度	0.895	0.829	0.821	0.851	0.846	0.824	0.846	0.797	0.807

③ 虽然待识文字的位移使得与它本身的标准文字的混合类似度值有所下降，但待识文字的位移同样会引起与极相似的候选文字的混合类似度值下降。因为极相似字集中的文字图形在位置上和结构上大部分是相同的，只是存在小部分的差异，所以当待识文字有位移时，与候选文字集中的相似字的类似度值同时下降这就保证了我们在区别极相似字时不发生错误的决策。表3为位移一网格时的类似度值。

表3

待识字 \ 类似度值 \ 候选字	且	旦
且	0.810	0.655
旦	0.660	0.811

待识字 \ 类似度值 \ 候选字	推	惟惟
推	0.878	0.735
惟	0.749	0.898

待识字 \ 类似度值 \ 候选字	闸	闸
闸	0.908	0.775
闸	0.775	0.863

以上的性能说明了用混合类似度来区别极相似字是可行的，区别结果可以说完全正确。

但混合类似度也有一定的不足，①占有较大的内存；②需要一个汉字模式库；③计算量较大。鉴于这些，一般在设计识别系统时，只存在很少的一部分字通过这一级。

五。结 论

本文对印刷汉字识别中出现的极相似字用类似度法进行了研究，并就字库中的3023个32×32的点模式汉字进行了相似字的统计工作。建立了相似汉字库。通过对库中相似字的识别，实验表明：对于识别最后阶段所出现的极相似字用本文所实现的混合类似度来区别是可行的。区别结果可以说完全正确。

另外，复合类似度也可用于识别阶段的分类。而相似字库的建立可以在研究识别方法中加以利用，验证其方法的好坏。给抽取汉字的特征提供一个检验的依据。

参 考 文 献

〔1〕 *Shoutch Hirai "Development of A Light pespormance Chinese Charactes Reades" Proc. 5th. IJCPR, PP 867~871, 1980.*

〔2〕 *Kenich Mori, "Advanse In Recognition of Chinese Character" Proc. 5th IJCPR, PP. 692~702, 1980.*

〔3〕 葛城纯夫"文字パタ－ン分布的要因分析上识别应用"〈昭和45年电气四学连合大会〉2816。

〔4〕 *Kunio Sakai "An Optical Chinese Churactes Reades" Proc. 3th. IJCPR. PP. 122~126. 1976.*

〔5〕 武裕朴、赵景台、杨力"2500个印刷体汉字识别研究"自动化学报 1984年1月。

〔6〕 杨力、梁刚、武裕朴"印刷体汉字粗分和识别研究"第四届模式识别论文集 1984年11月。

以笔画结构分析为基础的一种限制性手写体汉字识别方法（概要）

北京工业大学　无线电系

汪庆宝　　张　征　　刘鉴平

摘　要

本文介绍一种限制性手写体汉字的识别方法。这种方法从区别不同汉字的特征信息主要蕴涵在其笔画结构中出发，着重于提取和分析汉字的笔画，及其互相间的主要结构关系，采用九种笔画基元，结合最稳定的结构关系，对模式作出描述并识别。通过对频度最高的五百字进行的模拟试验，得出令人满意的结果。

一、引　言

汉字通常的表现形式可分为手写体与印刷体两大类。后者的图形模式结构规整，变化较少，相对来说较为容易识别，在国际范围内（主要是使用汉字较多的日本），大体认为识别方法基本趋于成熟，而且已有工业产品问世。对手写体汉字的识别（主要指脱机式），由于手写的任意性和汉字字种繁多复杂，识别的难度自然高得多。因此，虽然手写体汉字识别的研究工作，至今已开展十余年，但还没有得出公认较为良好的方法。

作者近几年研究了一种方法，从分析汉字笔画结构出发，采用九种基本笔画基元和几种结构关系描述被识别的汉字模式，据此进行识别，取得较好效果。

二、系统方法简述

汉字识别和其它模式的识别一样，首要的关键问题是选择特征。要找出反映区别不同汉字的本质特征，一方面固然需要观察、分析大量的汉字图形模式，然而另一方面，研究人类识字过程的机理，显然也是极为重要的。通过分析、研究、归纳可以发现，人们对汉字的识别是通过对汉字的一种描述实现的。这种描述以汉字的结构规律为基础，对于同一个汉字，尽管书写（或印刷）出来的汉字图形各有差异变化，但只要书写基本正常，人们总能对它作出正确的描述，从而完成识别的判决过程。

一般说，人们对汉字的描述方式是按照先部件（偏旁、部首、字根等）后笔画的顺序分层地进行的。按照这种方式，可以把汉字大体分为复合字与单体字两类。一个复合字由一定数量、一定类型的部件，按照一定的相对位置关系组成；而一个单体字或一个部件则由一定数量、一定类型的笔画，按照一定的相对位置和连接关系组成。不难发现，某些单体字既作为完整的汉字出现，同时也经常起部件的作用。例如“描”字，就是由提手“扌”、草头“艹”和“田”三个部件组成。而“田”既是一个单体字，也经常以部件出现在其它复合字中。这种分层的结构关系，可以用树形图表示，例如图1。

综上所述可以认为，一个汉字的本质特征就是组成这个汉字的部件和笔画的类型、数量、和它们之间的相对位置和互相连接的关系。当然，对于某些特殊的字，还应补充一些辅助的特征，例如，为了区别“士”与“土”、“未”与“末”等，就应补充对某些笔画长度特征的描述。

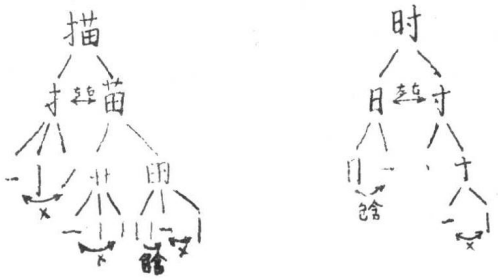

图1 汉字'描'和'时'的一种描述

上述方式虽然是一种结构性的描述，即它反映的是汉字的组成结构。这就表明，采用结构分析的方式进行汉字识别，特别是手写体汉字识别，是合理的。尽管统计识别方法具有处理过程简单、抗干扰性强等长处，却很难适应手写体汉字字种多、字形变化大的特点。

至此我们基本上可以设想，理想的汉字识别方案应当是依据被识别汉字的分层结构，将它分解为一个一个的部件，并记下其相互间的位置关系，然后又分解组成各部件的笔画，确定部件的类型，从而描述识别该字。显然，这一方案实质上是汉字生成的逆过程。

众所周知，评价一种识别方法或制订一种识别方案，不仅应考虑如何完整地体现被识模式的本质特征，而且还必须考虑实现它的技术的现实可行性。就目前来说，由于汉字部件互相嵌套结构的复杂性，分割提取这些部件的有效技术还是一个有待专门研究解决的课题。

基于对前人研究结果的综合分析可以表明，尽管描述汉字模式的分层结构方式比较理想，但实际上对于常用的汉字，往往包含不少冗余信息。因此，我们决定简化多层次的描述方式，而采用最低一层的笔画，及少量稳定的结构关系进行描述，从而构成了图2所示的识别系统方案。

三、基元及结构关系的选择

模式基元的选择除了考虑易于正确抽取外，还应同整个识别方法结合起来，使它的词义信息和结构关系适当相关连，能够充分地描述被识别的模式。

国内不少专家对汉字字形作了广泛的研究，为汉字模式基元的选择提供了可靠的依据。文献<4>为一种汉字字形编码方案规定了八种笔画，即横、竖、撇、捺、左折、右折、方、叉。这几种笔画基本符合我们对基元的要求，所以仅对它进行了少量修改和扩充，确定了九种汉字基元如下表所示：

基元名称	捺	竖	撇	横	竖钩	右折	左折	半框	框
笔　形	乀	丨	丿	一	亅	乛乙	𠃌𠃍	冂冂	口
代　码	1	2	3	4	5	6	7	8	9

上述基元除了半框和框以外，都是一笔写成的笔画。当然，半框和框实际上可以作为部件对待。不过由于它们在汉字中出现频繁，而且也较为容易提取（合成），因此也把它们算作基元。

描述手写汉字笔画基元之间的关系有：

·笔画顺序

·相对位置：　上、下、左、右

·相连：　端接（如“┌”、“└”）、交接（如“┬”、“├”）

·交叉：　如“×”、“＋”、“又”等

·包含：　如“回”、“日”等

在这些关系中，笔画顺序是非常有用的一种信息，但除了联机式的识别外，正确取得这一关系信息相当困难。通过初步的分析研究和试验，为了以尽可能节约的信息量实现我们的描述方式，我们选取了最后两种关系。因为它们是书写汉字时必须保持的结构，因而是可以取得的最稳定的关系信息。至于相连关系，则是极不稳定的，对于识别描述没有什么积极的贡献。

在选定模式基元并确定采用的关系信息以后，可以规定对被识别汉字书写的限制如下：

1）不添笔，不省笔；

2）应交叉的笔画必须交叉，不应交叉的笔画一定不能交叉；

3）不应连接的笔画避免连接，不能连笔。

这些限制在书写时不难满足，也不会给书写者造成过重的心理负担。

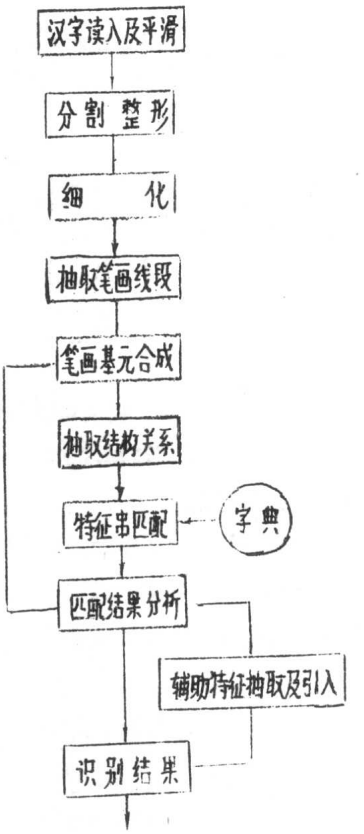

图2　识别系统结构流程

四、描述形式

在数字计算机中，用一维的代码串描述模式是最为简便的形式。我们定义描述汉字的代码串由三部分组成，即：

<交叉段><包含段><孤立笔画段>

交叉段描述一个汉字中互相交叉的笔画部分。其组成形式为

$$M[N_1\,Q_1\,[S_{11}\,S_{12}\cdots S_{1Q_1}]N_2\,Q_2\,[S_{21}\cdots S_{2Q_2}]\cdots N_M\,Q_M[\cdots]]$$

其中M为交叉块数、N 为第i个交叉块的交叉点数、Q 为组成该块的笔画数、S 为组成该块的笔画的代码。

交叉块可以按下述对交叉关系的定义抽取：

定义：　设$Q=\{q_1, q_2\cdots, q_n\}$是从被识汉字的骨架抽得的笔划集，则交叉关系是Q上的一个二元关系R，使得$(q_i, q_j)\in R$，当且仅当q_i和q_j中至少存在一个既不是q_i的端点，也不是q_j的端点的公共点。

计入关系R的传递性，可以很容易地实现交叉块的提取。

因为稳定的包含关系存在于“框”和其它笔画之间。当然，被包含的笔画之间也可能存在交叉关系。因此，我们规定包含段的形式为

$$C[D_1\,D_2\,[S_1\,S_2\cdots S_{D_2}]]$$

其中C为包含有其它笔画的框的数目、D_1是被包含的交叉块在交叉段中的序号，D_2=0说明没有被包含的交叉块；D_3为除交叉块以外的被包含笔画数、S则是这些笔画的代码。

包含关系的几何定义比较直观，这里不必赘述。

孤立笔画段描述不属于上述两种结构关系的笔画，其形式为

$$Y[S_1\ S_2 \cdots S_Y]$$

其中Y为孤立笔画数，S_i为笔画代码。

按照这种描述方式对几个汉字产生的描述代码串如下表所示：

字 例	代 码 串
国	1 1 2 2 4 1 1 3 4 4 9
京	0 0 6 1 1 3 4 5 9
汉	1 1 2 1 7 0 3 1 1 3

五、模拟试验及结果

本方案先后在以Dynabyte－Z80为主机和以IBM－PC/XT（或AT）为主机组成的实验系统上进行了模拟试验。

模拟试验是对汉字中频度最高的560个字中的500字进行的。实验所用字样由四人书写，将其中两个人写的字样（共700字）作为训练样本，取得所需的一些统计 值，而将其余两个人书写的一千字作为检验样本（零星试验未计入），试验结果如下表所示：

笔画抽取及合成	识别率	粗识率	误识率
一 次 性	93.1%	6.1%	0.8%
反馈或抽辅助特征	96.1%	3.1%	0.8%

对本识别方案所进行的试验。初步结果是令人满意的。通过对结果的分析也启示了我们对进一步工作的设想与安排。例如粗分类处理的加入（该工作已进行，未连入系统）、抽取特征方法的改进，笔画基元之间相对位置关系的引入以及硬件试验系统的改进等，都会对识别能力有进一步的改进和提高，而识别字种的扩大也正在进行之中。

参加本工作的还有周鸣方、唐前临、蒋小波、徐小平诸同志。

（参考文献及附图略）

点阵汉字的笔划分析及向量化算法

南开大学计算机与系统科学系
毛自强　朱耀庭

摘要：本文提出利用计算机上已有的汉字系统中的点阵字模汉字库，通过笔划分析，实现点阵汉字向量化〔1〕的方法。该算法有选择地搜索象点周围的八个邻点，实时地在笔绘图仪上自动输出笔划汉字，可广泛应用于计算机绘图和计算机辅助设计中。它所提出的思想和算法原则上适用手写体汉字的特征抽取和笔划分析。

关键字：向量化，字模，邻点，通道，搜索，模块。

§1 点阵字模与笔形分析

汉字点阵字模是由N×M象点矩阵构成。为分析方便，扩充此阵为（N+2）×（M+2）的矩阵，记为A，仍称其为象点矩阵。I行J列的象点用（I，J）表示，该点值记为A（I，J），它可取1或0。其中I＝0，1…N＋1；J＝0，1，…M＋1。

设定，A（K，J）=0 （K=0，N+1），A（I，L）=0（L=0，M+1）。其中I=0，1，…，N+1；J=0，1，…，M+1。

以下用P记扩充前象点矩阵中的一个任意的象点，其值记为A（P）。

定义一，A中P的八个相邻的点P_i（i=1，2，…，8）（如图1）称为P的邻点。P_i点相对于P点所在的方向称为P的第i个方向。

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

图1，P的8个邻点

定义二，若A中一串象点Q_1、Q_2、…Q_S（$S\geq 2$），$A(Q_i)=1$（i=1，2，…，S）且对所有$1<S$，Q_{i+1}是Q_i的邻点，则称Q_1，Q_S连成的折线为通道。

定义三，若象点P的值A（P）=1，且P的1，3，5，7方向上非零值的象点个数大于2，或P之2，4，6，8方向上象点值均非零，则称该点为交叉点。若P的所有方向上非零值的象点个数大于2则称其为交点。若P仅有一个方向有非零值象点，则称其为端点。

交叉点集合是交点的子集。

汉字点阵字模中，所有的通道构成了一个线图G，笔划汉字线图G'是G的子图。G的顶点由端点和交点组成，G的边是通道上相邻顶点间的通道。

汉字点阵向量化是把A中值为1的象点构成通道，且符合汉字的字形，需恰当选取G的边，形成G'。

对象点矩阵，从左到右，从上到下的搜索过程来搜索G'的边。由汉字笔形特征横（一），直（丨），撇（丿），点（、），折（乙，𠃌，乚）知，必须在搜索时考虑P点相对于前一点的方向，即笔划走向。表1列出了一种选择邻点的顺序规则。其中order(o)为进入P点的方向号，order(I)为第I个搜索方向上的方向号，其值为0时表示无第I个待搜索的方向，即对P的邻点的搜索结束。由于采用从左到右，从上到下的次序，故在对一通道搜索开始时，始终设进入P之方向为5（见图1）。

orde(o)	1	2	3	4	5	6	7	8
order(1)	6	6	7	8	1	2	0	4
order(2)	7	5	1	7	7	1	0	0
order(3)	0	7	6	1	8	4	0	0
order(4)	0	8	8	6	6	0	0	0
order(5)	0	0	2	0	2	0	0	0

表1 搜索P的邻点的方向号和顺序

由于G'是G的子线图，必须保证G'的完整性。故搜索过程中对搜索到的交叉点，起始点和终止点P，置A(P)=2可供再搜索。其它值为1的象点P,置其值A(P)=3，以保证再次搜索时，不再搜索到此点。为保证汉字笔划的合理性和完整性，又采用了两个方法弥补遗漏的细小线段，一个是对A的每行进行两次搜索；一个是通道回　搜索〔2〕，即搜索到一段通道终点后再回溯到最近的交叉点向另外的通道分枝搜索，最后回溯到起始点。

§.2通道搜索算法

设(PI，PJ)为当前的P点，(I'，J')为P的一邻点，A为扩充后的象点矩阵。算法如下：

1. I=0，JJ=0；
2. I=I+1，JJ=0，若I=N+1，则结束。
3. JJ=JJ+1，J=((JJ-1)(mod M))+1；若JJ=2M+1，则GOTO 2；
4. 若A(I，J)=1或2，则GOTO 5；否则，GOTO 3；
5. order(0)=5，PI=I，PJ=J，抬笔移到(PI，PJ)点，且落笔，记A(PI，PJ)=2；
6. 据order(0)的值确定搜索的邻点及其顺序；测试(PI，PJ)点，若其为交叉点，则记A(PI，PJ)=2；
7. 按已确定的邻点及其顺序搜索。若存在邻点(I'，J')使A(I'，J')=2，则落笔移到(I'，J')，且根据(I'，J')相对于(PI，PJ)的方向对order(0)重新赋值，记PI=I'，PJ=J'，A(I'，J')=3，GOTO 6；否则，GOTO 8；
8. 按已确定的邻点及其顺序搜索，若存在邻点(I'，J')使A(I'，J')=1，则落笔到(I'，J')且根据(I'，J')相对于(I，J)的方向对order(0)重新赋值，记PI=I'，PJ=J'，A(I'，J')=3，GOTO 6；否则记A(PI，PJ)=2，GOTO 3；

算法结束。

§3.通道回溯搜索算法

本算法可用于消除交叉点周围的断线现象。为了回溯必须记下搜索到的象点的坐标，及欲搜索的象点的邻点和其顺序。

数组PI(K)、PJ(K)、order(L，K)，分别表示搜索一段通道第K步时的象点坐标(PI(K)，PJ(K))，它的邻点和顺序(K=0，1，…；L=1，2，…，5)。order(o，K)为点(PI(K)，PJ(K))的进入方向号。算法中，当前点记为(PPI，PPJ)，其邻点记为(I'，J')。

算法如下

1. 令I=0，J=0；
2. I=I+1，J=0；若I=N+1，则结束。
3. J=J+1，若J=M+1，则GOTO 2；
4. 若A(I，J)=1，则GOTO 5；否则GOTO 3；
5. order(0，0)=5，PI(0)=I，PJ(0)=J，PPI=I，PPJ=J，K=0，抬笔移动到(I，J)后落笔，且记A(I，J)=2；

6.根据order(o,k)值确定搜索的邻点及其顺序；测试(PPI,PPJ)，若是交叉点，则记A(PPI,PPJ)=2；

7.按已确定的搜索邻点和顺序搜索，若存在邻点(I',J')，使A(I',J')=1，则落笔移动到(I',J')，置K=K+1，根据(I',J')相对于(PPI,PPJ)的方向决定新的order(O,K)的值，记PPI=I',PPJ=J',A(PPI,PPJ)=3,PI(K)=PPI, PJ(K)=PPJ,GOTO 6; 否则GOTO 8;

8.K=K-1，若K≤0则GOTO 3；否则，若A(PI(K),PJ(K))=2则抬笔到象点(PI(K),PJ(K))，且GOTO 7；否则GOTO 8;

算法结束

§4.点阵汉字向量化模块

将上述算法编成程序模块，用户只要指明要绘的汉字，其尺寸大小及在绘图面板上的坐标位置，便可完成这一向量化转换，且绘出汉字。我们已在IBM PC/XT微型机上，对CCBIOS汉字系统编制了该模块，结构如下。(详见〔4〕)

1.找汉字内码

2.由内码算出汉字点阵在内存中的地址

3.取出字模信息

4.用向量化算法把点阵汉字向量化

§5.结束语

通过实践，我们认为在应用软件中调用这样的向量化模块是很方便的，平均写出一个汉字仅需15秒钟，更重要的是作到了一个字库两用，节省存贮，节省盘空间。该算法尽管是在单线体上进行的，但对于多线体的点阵汉字以及一般手写体汉字(在进行了骨架抽取后)也是适用的。

我们在工作中，始终得到了张朝池副教授的耐心指导，在此表示衷心的感谢。

参考文献：

〔1〕Glbson,L.D.lucas,"Vectorlzatlon of Rastor Uslng Hlerachlcal Methods",CGIP 20(1)Sept.1982.P82-89.

〔2〕何华灿"人工智能导论"航空专业教材编审组，1983.

〔3〕电子工业部第六研究所"CCDOS(CCBIOS)V2./2.1使用指南及其附录"，1984.6.

〔4〕毛自强，朱耀庭，"由点阵汉字绘制笔划汉字的程序设计"，1986.1.

限制性手写体汉字的一种平滑和细化方法

北京工业大学无线电系

蔡小波　王好博

摘　要

汉字的预处理和细化对汉字识别中的识别率和识别速度的提高均有重大的意义。本文在考虑了汉字的结构特点和处理速度后，提出了汉字平滑和细化的一种方法，通过对摄象机读入的1500个限制体手写体汉字的处理。可以看到该算法与现有的细化算法相比，具有文字失真小，处理速度快的特点。

(一) 引　言

在计算机汉字识别中，首先是光电转换。如果书写所用的笔质量不好，文字背景不小心蹭上了油墨，光电转换装置本身的噪声等等因素可能使得光电转换后的文字笔划凸凹不平，在背景中出现污点或在文字笔划中出现空白凹陷，如不经平滑，将严重影响文字的进一步处理。文字笔划的粗细一般不包含文字的信息，所以光电转换后的文字一般是要进行细化，即抽取汉字的骨架，这样可大大减轻后级处理量。平滑的主要目的也就是为了保证细化的质量。我们在这些方面进行了一定的研究工作，实验证明效果很好。

(二) 平　滑

平滑就是去除文字笔划边缘突出的毛刺（包括去除孤立点）和补上凹陷的小空缺，使笔划平整光滑，以利细化。某一个点 z_0 是否毛刺或凹陷，可以通过对出现在如图1所示的3×3窗口里的图象元素使用明确的逻辑规则来判定。观察从光电转换装置输入到计算机中来的大量文字，我们规定：一个黑点如果出现连续环绕着它的五个点都是白点，而其余三个点满足一定的条件，则判 z_0 是毛刺。三个点要满足的条件，我们是考虑以下的情形（见图2）。对于(d)，删去 z_0 可能破坏笔划的连通性，对(e)，z_0 可能是端点。而对于(a)、(b)、(c)三种情形，绝大多数均是毛刺，应该删去。将图2旋转90°的整数倍，则包含了3×3窗口的所有情形。平滑的另一个任务是填补小凹陷。如果环绕某个白点 z_0 的五个点都是黑点，或者 z_1，z_3，z_5，z_7 都是黑点，则判 z_0 是小凹陷。通过观察1500个汉字的细化情况，证明上述的平滑处理是适当的，基本上去除了对细化影响较大的噪声。由图8可以看到平滑对细化的影响。

如果我们以“1”来表示黑点，以“0”来表示白点，判定某个点是毛刺时，就把它的值由“1”改为“0”，当判定某个点是凹陷时，就把它的值由“0”改为“1”。在编写计算机程序时，我们发现，如果以从左到右，从上到下的扫描平滑文字，前面的点平滑后的结果会影响到后面点的平滑。特别是在笔划很多的汉字中，两个笔划间的距离往往很小，甚至间隔只有一比特，在这种情况下，如果左面的点（或上面的点，记为 t_1）被判成了凹陷，将它平滑掉后，扫描到它右面的点（或下面的点，记为 t_2）时，由于 t_1 已改为黑点，则 t_2 仍会被判为凹陷。同理，t_2 右面的点（或下面的点）均有可能被判成凹陷，平滑结果使间隔相近的两笔划变成了一粗笔划。对于毛刺的平滑，也有类似的问题。这种情况是必须避免的，所以我们采用并行处理，即对所有的窗口均使用文字的原始数据。对前面已经平滑过的点，暂时先不改动，仅仅作个记号，当不再需要这些点时，再将这些点改动。由于平滑后的文字接下来就是细化，故作过记号的点可以在细化的第一次扫描中改动。而平滑仅仅需要对文字图象扫描一次，处理时间很短。

(三) 细　　化

细化与平滑处理有某些相似之处。细化一般应用剥离技术，根据笔划的粗细，一个字一般要经过若干次的剥离，即若干次迭代来达到细化的目的。对细化的要求是：

(1) 保证细化后文字图象的连通性。

(2) 文字图象的骨架形状要保留，细化结果应尽量是原来文字的中心线。

(3) 处理时间要短。

我们仍然采用图1所示的3×3窗口，z_i（$i=1\sim7$）与z_0是8连通的。在文字图象中，只要两点间是8连通的，我们就称该两点是连通的。如果一个点的删除不影响文字的连通性，我们就称该点为8－可删点。连通数

$$N_{08}=\sum_{k\in s_1}(\bar{z}_k-\bar{z}_k\cdot\bar{z}_{k+1}\cdot\bar{z}_{k+2})$$

其中　$S_1=\{z_1,z_3,z_5,z_7\}$

反映了一个点与窗口中其余8个点的连通性。可以证明一个点是8－可删点的充要条件是$N_{08}=1$。在$N_{08}=1$的情况下，根据汉字结构特点，属于不可删去的仅有两种情况，如图4所示。在(a)中，显然z_0是端点，因此是不可删除的。在(b)中，虽然删除z_0也不破坏连通性，但根据汉字的结构特点，“T”型结构是很多的，删除这样的点会破坏汉字的骨架特征，故应该保留。对(a)，(b)两情况的结论也适合于它们的90°整数倍旋转的情况。

为了使细化结果尽量是原始文字的中心线，我们对文字图象采用四个方向的扫描，如表1所示。在经过较为理想的平滑后，不论哪个方向的扫描，首先遇到的黑点一定是边界上的点。比如在第一个方向的扫描中，我们先判断该点上、下两点是否为“0”，“1”，如是，我们就确定此点为边界上的点，如果它再满足上述所说的可删条件，那么它就是可删点。但是我们并不马上把它删去，因为这样会在一次扫描完成时，将所有笔划均变成一比特宽的单线。而在经过一个周期即四个方向的扫描后，一个字在四个方向上均被剥“光”，这样的情况对横的笔划来说固然是好的，但就其它方向笔划来说，将是不允许的。所以我们仍然与平滑时一样在每一方向的扫描中采用并行处理。在一次扫描过程中，在被判断应予以删除的那些点，均在其源位置上做一标记，以区别于“0”和“1”，而不是马上删除，这样，在扫描其下面的点时，就不可能再出现“011”或它的90°整数倍旋转这样的结构，因为此时的零是做了标记的，而不是“真正”的零。因此，在一次扫描中，就不会出现上面说到的那种一次全剥“光”的现象。而那些做过标记的可删点，在下一个方向的扫描中予以删除，这样做使计算机费时达到最小。本算法在一个周期即四个方向的扫描结束时，一个字就在四个方向上均剥掉一层，这时需要判断一下，整个图象中是否还有可删点，若有则仍需重复扫描，直至笔划宽度为一比特为止。本算法包括平滑在内共需对文字图象扫描4N＋1次（N为剥离层数）。实验中，我们分别用8086汇编语言和IBM－PC BASIC编译语言编写程序，对于取样为64×64的文字图象（其笔划粗细在4比特以下），前者处理时间为1秒左右，后者为2～3秒。从细化结果看，基本上达到上述几个要求。细化后的文字如图5所示。

(四) 致　　谢

本文工作是在汪庆宝导师的精心指导下进行的，特此感谢。

(五) 主要参考书

1. THEO PAVLIDS, ALGORITHMS FOR GRAPHICS AND IMAGE PROCESSING, 1981

2. Y. F. TSAO AND K. S. Fu, PARALLEL THINNING OPERATIONS FOR DIGITAL BINARY IMAGES, IEEE Comp. Society Cont. on P.R and I.P, 1981

3. 陈尚勤，魏鸿骏. 模式识别. 人民邮电出版社. 第八章

4. F.W.M. STENTIFORD and .R.G. MORTIMER, Some New Heuristics for Thinning Binary Handprinted Characters for OCR. IEEE TRANSACTIONS ON SYSTEM, MAN, AND CYBERNETICS. Vol. SMC-13, №1, JANUARY/FEBRUARY 1983

表 1

编号	细化方向	扫描点移动方向
1	上→下	左→右
2	左→右	下→上
3	下→上	右→左
4	右→左	上→下

x_4	x_3	x_2
x_5	x_0	x_1
x_6	x_7	x_8

图1 3×3窗口

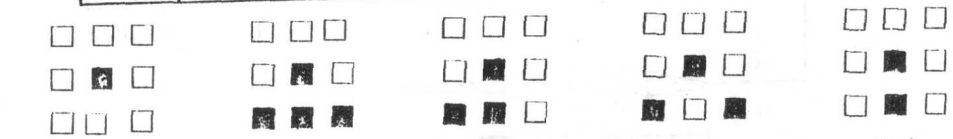

图2 (a) (b) (c) (d) (e)

图3 (a)原文字 (b)无平滑细化 (c)有平滑细化

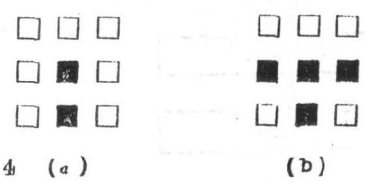

图4 (a) (b)

图5 细化后的文字 (a) (b)

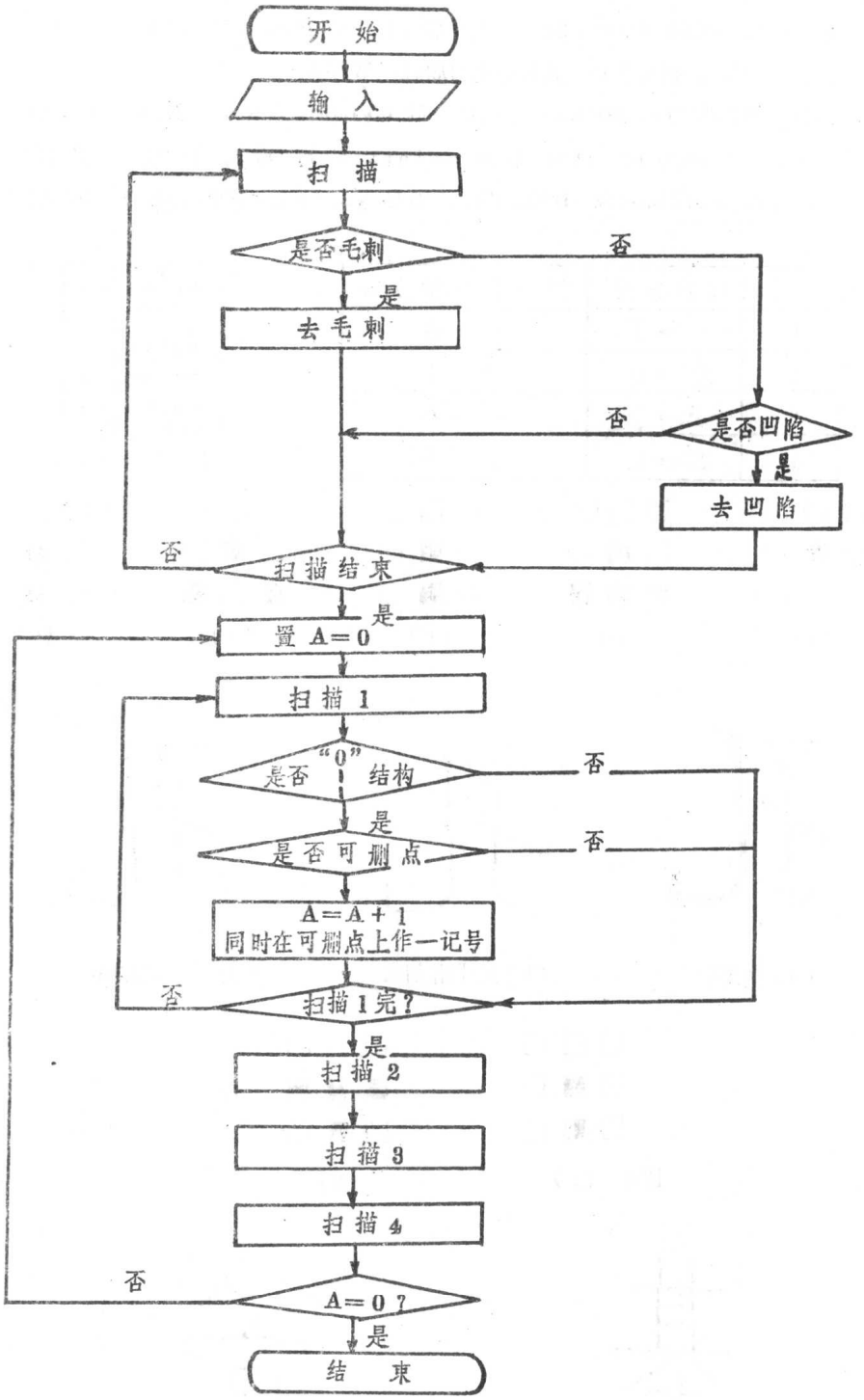

表2　程序框图